

Watertight Anisotropic Surface Meshing Using Quadrilateral Patches

Robert Haines
Massachusetts Institute of Technology
haines@mit.edu
and
Michael J. Aftosmis
NASA Ames Research Center
aftosmis@nas.nasa.gov

Abstract

This paper presents a simple technique for generating anisotropic surface triangulations using unstructured quadrilaterals when the CAD entity can be mapped to a logical rectangle. Watertightness and geometric quality measures are maintained and are consistent with the CAPRI default tessellator. These triangulations can match user specified criteria for chord-height tolerance, neighbor triangle dihedral angle, and maximum triangle side length. This discrete representation has *hooks* back to the owning geometry and therefore can be used in conjunction with these entities to allow for easy enhancement or modification of the tessellation suitable for grid generation or other downstream applications.

Introduction

The premier design goal for the Computational Analysis **P**rogramming Interface (CAPRI) [1] is that geometry access be appropriate for CAE developers. The complete Computational Geometry (CG) perspective is avoided while maintaining full functionality. Early in the design and implementation of CAPRI, it became obvious that providing an Application Programming Interface (API) that only gives the programmer access to the geometry and topology of a solid part was insufficient. The burden of deciphering the CAD data and attempting to generate a discrete representation of the surfaces required for mesh generation was too great. Fortunately, many grid generation systems (used in CFD and other disciplines) can use STL (*Stereo Lithography*) files as input. Combining a discretized view of the solid part as well as its geometry and topology can provide a complete, and easier to use, access point into the CAD data. A tessellation of the object that contains not only the mesh coordinates and supporting triangle indices but other data, such as the underlying CAD surface parameters (for each point), as well as the connectivity of the triangles, assists in traversing through and dissecting the CAD representation of a part.

An important aspect of CAPRI is that it provides CAD vendor neutral access to

all of the data obtained from the models that is to be passed back to the application. The triangulation generated by CAPRI is guaranteed to be watertight, regardless of the CAD kernel in use. Some CAD system geometry kernels can provide data of this quality (i.e., UniGraphics, Parasolid, CATIA and ComputerVision). Other CAD systems can provide the data, but it is not of sufficiently high quality to use. (For example, Pro/Engineer requires one to buy Pro/MESH to get a closed triangulation.) Finally, SDRC's Open I-DEAS API does not provide access to a triangulation at all. The fact that not all CAD systems provide such a tessellation has forced the development of a surface triangulator within CAPRI for CAD solid parts that *does* meet all of the quality requirements.

It should be noted that CAPRI's tessellations are not intended as the starting point for computational analysis (though they could be used in some cases). CAPRI sees only geometry, and it cannot anticipate the smoothness, resolution or other requirements of the downstream application(s). The triangulations approximate the geometry only; some processing of the tessellation is expected in order to refine the triangulation to a state suitable for the physical problem being investigated. The triangulation can be enhanced through either physical or parameter space manipulation, using point "snap" and (u, v) surface evaluations routines provided by the CAPRI API [2]. The triangulation technique used within CAPRI displays the following characteristics [3]:

- Robust. It is imperative that the scheme work for all possible topologies and provide a tessellation that can be used.
- Correct. The triangulation is of no use if it is not true to the CAD model. The tessellation must be logically correct; i.e. provide a valid triangulation in the parameter space (u, v) of the individual surface. It must also be geometrically correct; i.e. depict a surface triangulation that truly approximates the geometry. This involves ensuring all facets have a consistent orientation with no creases or abrupt changes in triangle normals. Correctness in both physical and parameter space allows CAPRI based application enhancement schemes to operate in either or both.
- Adjustable. To minimize the post-processing of CAPRI's tessellation for a specific discipline or analysis, some a priori adjustment of the resultant quality is available. It must be noted however, that any criteria may not be met (especially near the bounds of a CAD object) due to issues of closure and solid model accuracy. This goal may conflict with the more important characteristic of being watertight and having a smooth surface representation. The parameters are:
 - Maximum triangle side length. Any triangle sides (not on a CAD Edge) longer (in (x, y, z)) than a specified value are bisected.
 - Maximum dihedral angle between two triangles. Any two triangles on the same CAD Face whose (x, y, z) facet normals differ by more than the input value will be broken up.

- Chord-height tolerance. When the deviation between triangle center and actual surface (in (x, y, z)) is greater than the specified value then the triangle is subdivided by inserting the center point.
- No geometric translation. To truly facilitate hands-off grid generation, anything that requires user intervention must be avoided. All data maintained within CAPRI is consistent with the CAD's solid model representation. An alternate or translated representation is not used, because then the result will be something different than resides within CAD.
- Watertight. Triangulated CAD solids are closed and conformal; having this characteristic allows for meshing without "fixing" geometry. For the tessellation of a solid object, this means that all Edge (trimming) curves terminate at consistent coordinates of the bounding Nodes and a single discretization for Edge curves be used on both surfaces sharing the common Edge. Each triangle side in the tessellation is shared by exactly two triangles, and the star of each vertex is surrounded and bounded by a single closed loop of sides. The triangulation is everywhere locally manifold. In a manifold triangulation, there are no voids, cracks or overlaps of any triangles that make up the solid.

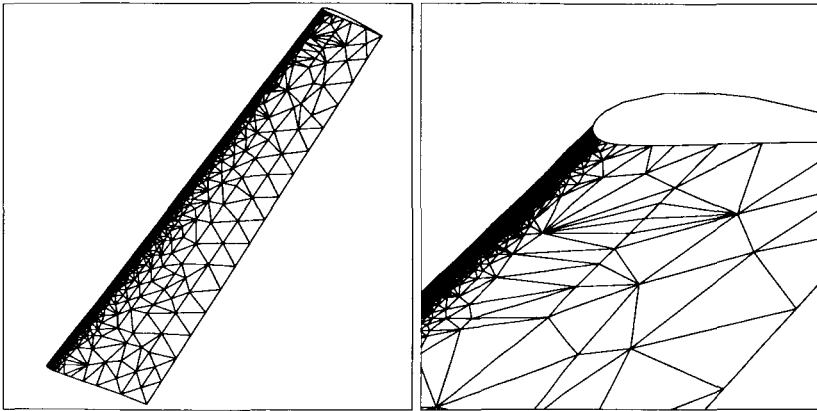


Figure 1. Isotropic Triangulation using default Tessellation scheme. # points = 6424, # triangles = 12639, Total CPU time = 3.8 sec.

As can be seen in Figure 1 the dihedral angle based refinement is sensitive to anisotropic curvature in the surface. Since the simple subdivision rules do not attempt to align inserted vertices with the direction of principle curvature, surface bends, cylindrical trailing edges, fillets and a surface with any linear feature require a large number of triangles to satisfy the angle metric. When combined with the isotropic nature of the MinMax triangulation in (u, v) , this misalignment can make dihedral-angle based refinement expensive and provide results with high counts.

The CPU time to generate the complete tessellation of the solid (4 Faces) is 3.8 sec¹. A careful timing analysis indicates that as the number of triangles increases on a Face, the proportion of time spent swapping for surface recovery grows rapidly. While triangulation speed has been improved through the use of recursive swaps, many triangles will still be required at anisotropic surface features without a more sophisticated site insertion strategy [3].

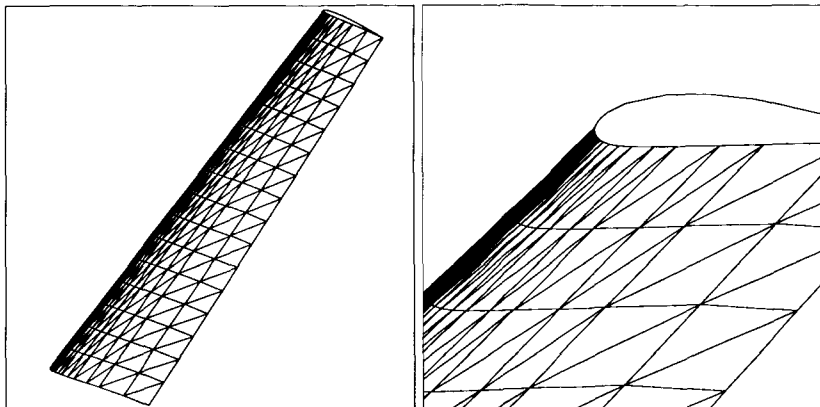


Figure 2. Triangulation based on a Trans-Finite Interpolation scheme. # points = 340, # triangles = 608, Total CPU time = 0.1 sec.

In an attempt to mitigate this problem a simple Trans-Finite Interpolation (TFI) scheme was implemented. The TFI procedure takes the (u, v) s along the bounds of the quadrilateral face and interpolates (u, v) s to interior points. These new parameter pairs can be used to evaluate to physical coordinates and therefore simply (and quickly) fill the Face with sub-quadrilaterals. These cells can be further subdivided to produce a triangular mesh. This initial scheme has the following restrictions:

1. Face must have only one bounding Loop (i.e. no holes)
2. The Loop must contain 4 Edges
3. Number of points found in opposing Edges must match

These limitations ensure few Faces can use this algorithm. But, the results are both enlightening and encouraging for Faces (with linear features) when the scheme can be employed. This can clearly be seen in Figure 2 where the resultant mesh is more regular, the counts are significantly reduced, the time to generate is a small fraction of the original, while satisfying the same geometric quality metrics. It should also be noted that the triangles generated are far from isotropic in particular those at the leading edge of the wing.

¹All timings in this paper are generated on a 1.8GHz Pentium 4m running LINUX.

Another less obvious advantage of this type of scheme is consistency. Any change in geometry will produce an entirely different triangulation using the standard scheme. The topology of the TFI mesh is driven only by the bounds of the quadrilateral Face. Therefore, if the point count at the Edges remain constant then the interior triangulation is consistent. This can be useful in design settings when differencing is employed to determine parameter sensitivities. This is because point movement within a Face can be tracked.

Unstructured Quadrilateral Patch Fill

The easiest way to ensure a watertight triangulation of a solid is to first discretize the Edges that bound each Face. Face tessellations can then be performed using the Edge points and filling in the interior without regard to the neighboring Faces. In an attempt to capture more Faces using the TFI scheme it is obvious that the restrictions need to be relaxed.

Since we do not wish to change the manner in which the general triangulation scheme is done, it would helpful to find a TFI-like method that does not have restriction #3. This method must also be able to produce near-normal sub-quadrilateral elements near the bounds of the Face in (u, v) so that linear features (found at the Edges) can propagate into the Face in an anisotropic manner.

One Set of Opposite Sides Match

In this simplest case, one set of opposing quadrilateral sides match in point count – the other does not. If we assume that the largest of the mismatching sides is found at the right then the blocking that can be used to subdivide the Face is seen in Figure 3.

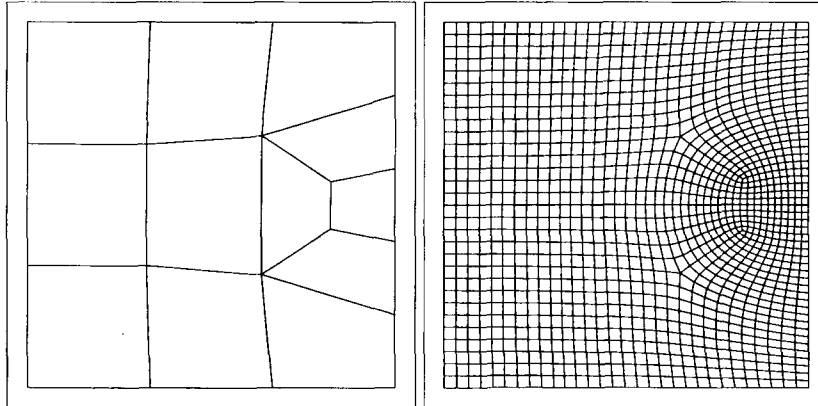


Figure 3. Generic block template and example of one set of opposite sides having the same point count.

By examining the block template (the left side of Figure 3) one can see that the additional points in the larger side are connected back to themselves by making a loop of elements. This loop is brought back to about $1/3$ of the way in the opposite direction so that these elements do not penetrate too far to the left. Also the turning of the loop does not end up too close to the generating side so that the quads at the right side can be close to normal. The picture seen on the right-hand side of Figure 3 is a completed mesh using this block template. The size of each block is determined by either $1/3$ of the appropriate side count or $1/2$ of the difference between the opposing sides. Note that an odd difference is made even (by reducing the count by 1). This point is placed back into the final mesh by subdividing the appropriate sub-quad into 3 triangles instead of 2.

After each sub-block is populated, the result is improved by applying a Laplacian smoother.

It should be noted that the constraint of the side with the largest count being on the right is one of convenience. A set of up to 3 90° rotations can take any configuration and place in the above situation.

Opposite Sides Differ by a Constant

Another simple case to consider is when opposing sides differ by the same count. The blocking can be found in the left-hand picture of Figure 4. Here we assume that the largest side is on the right and the largest side from the 2 others can be found at the bottom. Again this constraint is artificial. A set of up to 3 90° rotations and a reflection can take any configuration and place in this setting.

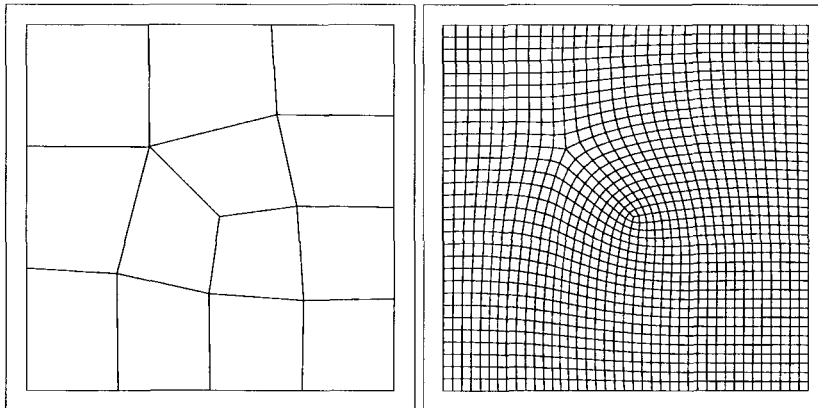


Figure 4. Generic block template and example of where opposite sides differ by the same point count.

For this case, it can be seen that the additional points generate elements that loop from the bottom and end up at the right quadrilateral side. The picture seen on

the right-hand side of Figure 4 is a completed mesh using this block template and is constructed in a similar manner to the first case.

General Case

The blocking for the general case can be found in the left-hand picture of Figure 5. This is, in fact, the combination of both of the simpler cases described above.

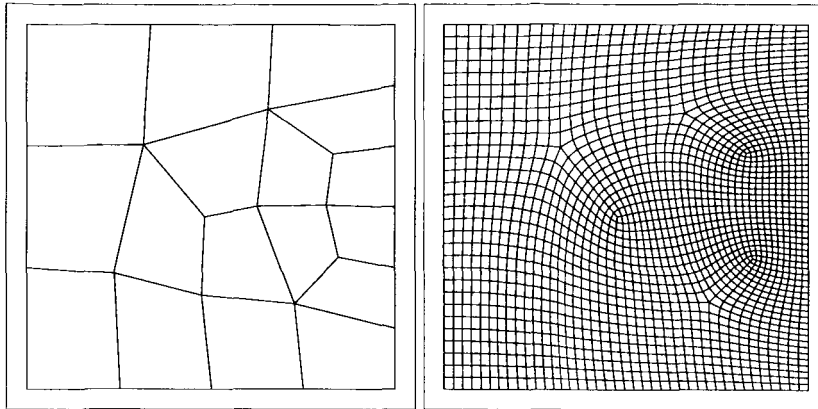


Figure 5. Generic block template and example of the General case.

As in the last case, we assume that the largest side is on the right and the largest side from the 2 others can be found at the bottom. This is not a constraint; a set of up to $3\ 90^\circ$ rotations and a reflection can take any 4 discretized sides and place in this configuration.

17 blocks are required in order to subdivide the original quadrilateral and both simpler cases can be seen imprinted in the blocking. The circular loop is obvious on the right as depicted in Figure 3 and the set of elements coming up from the bottom can still be tracked to the right of the original quadrilateral. Here it is clear that the elements get further broken up when the circular loop intersects this group of elements.

The picture on the right-hand side of Figure 5 is a completed mesh using this block template and is constructed in a similar manner to the first case; fill each block and then apply a Laplacian smoother. One can clearly see that local orthogonality has been maintained and those places that deviate from normal to the Face sides are far from those sides.

This scheme can deal with any 4 sides discretized with any number of points except for these conditions:

- A side has less than 4 points. This is because the basic method requires at least 3 blocks on a side unless the true TFI algorithm can be applied.

- The number of points on opposing sides differs greatly. It is possible to have situations where this scheme does not reduce the vertex/triangle count over the default triangulation. This occurs when there is a great disparity between opposing side counts. It has been found that when the side vertex count ratio (largest/smallest) gets greater than 3 the benefit begins to be minimized. This heuristic is used to limit the use of the anisotropic scheme.

Loops that do not have 4 Edges

In an attempt to further remove the constraints of this TFI-like anisotropic triangulation scheme we now look at restriction #2:

3 Edges

Under the limited set of circumstances that a Face with 3 Edges contains a degenerate Node; the Face can be viewed as a quadrilateral and the technique described in the previous section can be used. Degenerate mappings can at the tip of a conical surface and the pole of a spherical surface. This can be identified by a discontinuity in the values of (u, v) on either side of the Node.

When this situation is found, the following technique can be used:

- Create a virtual side at the degenerate Node. A new side is created with the same number of vertices as the side that is now opposing. The (u, v) s copied from the opposite side and appropriate parameter (the one not incrementing/decrementing) is set to that found with the Node.
- Perform TFI or **One Set of Opposite Sides Match** scheme.
- Deflate the virtual side. When the sub-element quadrilaterals are broken up, any triangles that have an edge on the virtual side are not included in the final tessellation.

More than 4 Edges

If one can determine sets of Edges that are part of a larger continuous curve in physical space, these can be considered a single quadrilateral side. If after analyzing all Edges there are 4 sides then the schemes described in this paper can be applied.

The technique used to examine each pair of Edges is simple and requires the following to be true:

- Is the Edge an isocline (have roughly a constant u or constant v)?
- Is the pair the same type of isocline? If so, then the pair can be considered part of a single quadrilateral side.

Example – A Turbine Blade

The following question needs to be asked; can the methods outlined in this paper show real benefits for actual CAD parts? To answer this question we will use an example from turbomachinery. The turbine blade part contains not only the aerodynamic shape but also the hub and tip casements including the *fir tree*. The full geometry (generated in Pro/ENGINEER) can be seen in the left-hand picture of Figure 6 and a blow-up of the hub/leading-edge region can be seen on the right.

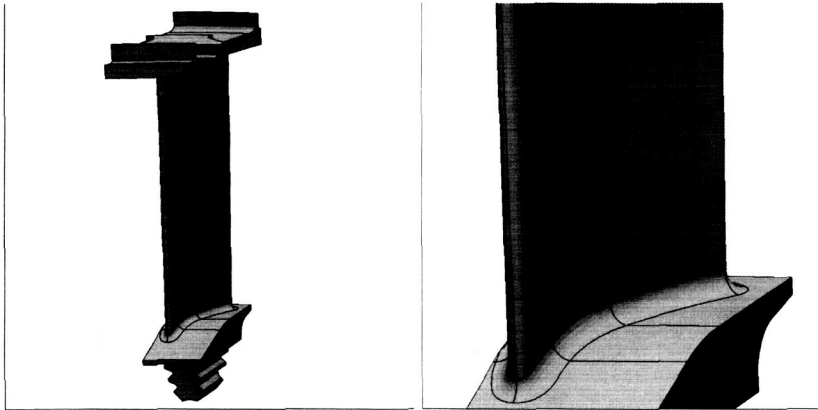


Figure 6. A turbine blade CAD part on the left. There are 94 Faces that represent the solid. On the right is a blow-up of the blade leading-edge hub junction where the fillets can be seen.

Figure 6 shows that the fillet between the hub and the aerodynamic shape is broken up into quadrilateral patches (this is seen with most all CAD systems). The set of trimming curves along the upper bounds of the fillets is a single entity as far as the aero shape is concerned (broken up to maintain the manifold aspect of the solid). It should also be noted that the aero shape is split at the leading edge into suction and pressure surfaces each reflected in a single CAD Face.

The left-hand side of Figure 7 shows the isotropic triangulation of the blade surfaces and the fillets for the hub blown-up view. The entire tessellation of the solid contains 66308 triangles and took 60.1 CPU seconds. On the right one can see the triangulation of the same Faces (all using the anisotropic scheme). The complete solid contains 22262 triangles and took only 2.53 CPU seconds being able to treat 56 of the 94 Faces as quadrilateral patches. From the performance improvement one can assume that the Faces that consumed most of the time for the isotropic triangulation were handled by the quadrilateral scheme (the suction and pressure surfaces).

The pressure surface is properly handled even though it is bounded by more

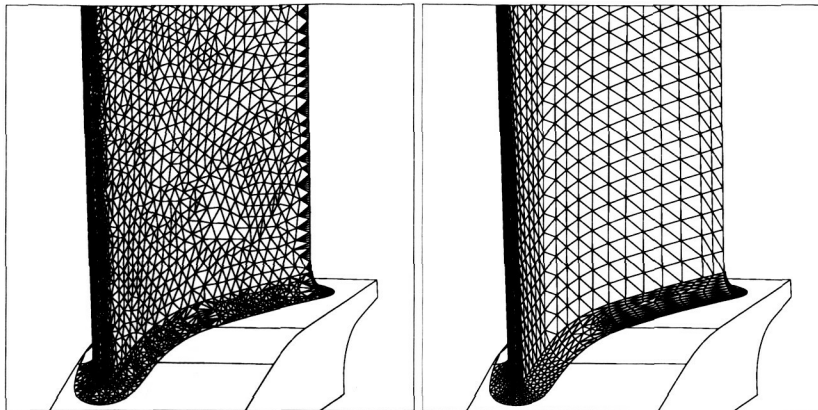


Figure 7. The surface mesh displayed on the Blade surface and some of the fillets. The picture on the left displays the results from the isotropic tessellation scheme. Seen on the right are the same Faces triangulated with the anisotropic method.

than 4 Edges (3 Edges can be seen at the mating with the fillet Faces). An abrupt spacing change can be noted in this Face's triangulation and looks odd at first inspection. The location of this change is at the position where the fillet Faces are subdivided. Remember that the interior points of a TFI algorithm are driven by the spacing at the frame. In this case each Edge has been discretized separately and there is much more curvature near the leading edge producing a finer set of points. The fillet Edge interior to the leading edge sees much less curvature and hence displays a coarser spacing.

It is easy to imagine that both the performance gain and the ratio of quadrilateral Faces to total CAD Faces to be high in this example. In the area of performance this may be true; if the suction and pressure surfaces of the blade had cooling holes neither of the Faces could have used the anisotropic method. But, after a (non-rigorous) survey of CAD parts a ratio of around 50% appears to be average.

Discussion

The anisotropic quadrilateral patching method described in this paper has been integrated into CAPRI, but it is not fully automatic. This is due to the possible situation where the quadrilateral approach cannot be applied and the default triangulation method must be used. Due to the isotropic nature (in (u, v)) of the tessellation scheme the Edge discretization must be finer in regions of high surface curvature so the geometry can be captured. That is the default. So without (hands-on) intervention fewer CAD Faces employ the quadrilateral scheme because the constraint based on **the number of points on opposing sides**

differ greatly becomes invoked.

Effort is underway to integrate the two surface meshing schemes so that it can become fully automatic. This is difficult because the Edge discretization is done before the Faces are tessellated (a requirement of the watertight attribute). The last phase of the Edge discretization is the examination of the local curvature of both of the Faces that touch the Edge. If curvature is found then the Edge tessellation continues to be enhanced. This is not only not necessary for the quadrilateral method, but can significantly reduce its quality if employed.

This also brings up the possibility of taking the current default surface triangulation and supporting anisotropic meshing. One of the swapping techniques used in the isotropic triangulation scheme drives the tessellation toward Delauney (Min-Max). This is done in the underlying surface's parameter space (u, v) . Since the parameter space is artificial (i.e. not physical) it could actually be any 2D mapping. Therefore a transformation from a 2D space that could support an anisotropic stretching to the surfaces parameter space would be all that is required to achieve the anisotropy found in the quadrilateral patch method. One could base the transform on the spacing found at the (u, v) rectangular bounds for the Face.

Finally, if this is successful, then the last phase of the Edge discretization may be able to be removed. Or, could be modified so that the spacing reflects the transformed 2D space. This then could mitigate restriction #1.

References

- [1] R. Haimes, and G. Follen. **Computational Analysis PRogramming Interface**. Proceedings of the 6th International Conference on Numerical Grid Generation in Computational Field Simulations. University of Greenwich, June 1998.
- [2] M. Aftosmis, M. Delanaye and R. Haimes. Automatic Generation of CFD-Ready Surface Triangulations from CAD Geometry. AIAA Paper 99-0776, January 1999.
- [3] R. Haimes, and M. Aftosmis. On Generating High Quality "Water-tight" Triangulations Directly from CAD. Proceedings of the 8th International Conference on Numerical Grid Generation in Computational Field Simulations. Honolulu HI, June 2002.